

What should be used for Reconfigurable HPC, FPGA or Coarser-Grain Reconfigurable Architecture?

Kentaro Sano

Leader, Processor Research Team

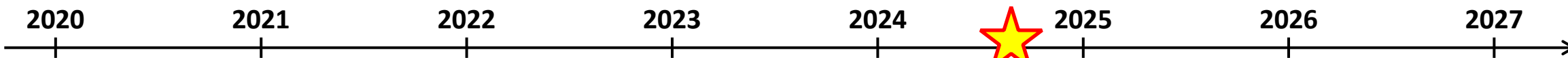
Leader, Advanced AI Device Development Unit

Leader, Architecture Research G in Feasibility Study for FugakuNEXT

RIKEN Center for Computational Science (R-CCS)

Processor Research Team, Advanced AI Device Development Unit

Goal: Establish HPC & AI architectures suitable in Post-Moore Era



1. Advancement of Fugaku

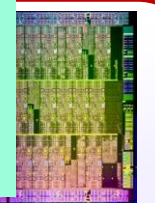
FPGA

- ✓ Research on **Functional extension with FPGAs** (FPGA cluster development, specialized hardware for HPC)

2. Exploration of new HPC & AI architectures

- ✓ Research on reconfigurable accelerator (e.g. **CGRA**)
- ✓ Research on next-generation **AI chip architecture**

General purpose computing and AI

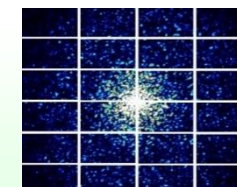


Special purpose computing

FPGA

3. Near-sensor processing / Scientific edge-computing

- ✓ FPGA-based processing for **X-ray imaging detector** (RIKEN Spring-8)
- ✓ Data-compression hardware for edge-computing (ANL)



FPGA

4. Backend of Fault-Tolerant Quantum Computers

- ✓ Specialized hardware for **quantum error correction** (Hardware algorithms, FPGA demo targeting RIKEN quantum device)

This Talk

- What should be used for Reconfigurable HPC, **FPGA or CGRA?**
- **FPGA-based HPC**
 - ✓ **ESSPER** : Elastic and scalable **FPGA-cluster** system for high-performance reconfigurable computing, as Prototype FPGA cluster for HPC
 - ✓ FPGA-based decoder for **quantum error correction** (in progress)
- **CGRA** (Coarse-grained reconfigurable array) for HPC
 - ✓ **RIKEN CGRA** research
 - ✓ Architectural exploration example for HPC workloads

Problem : System Power Consumption

Average power consumption

- ✓ in TOP10, TOP50, TOP500

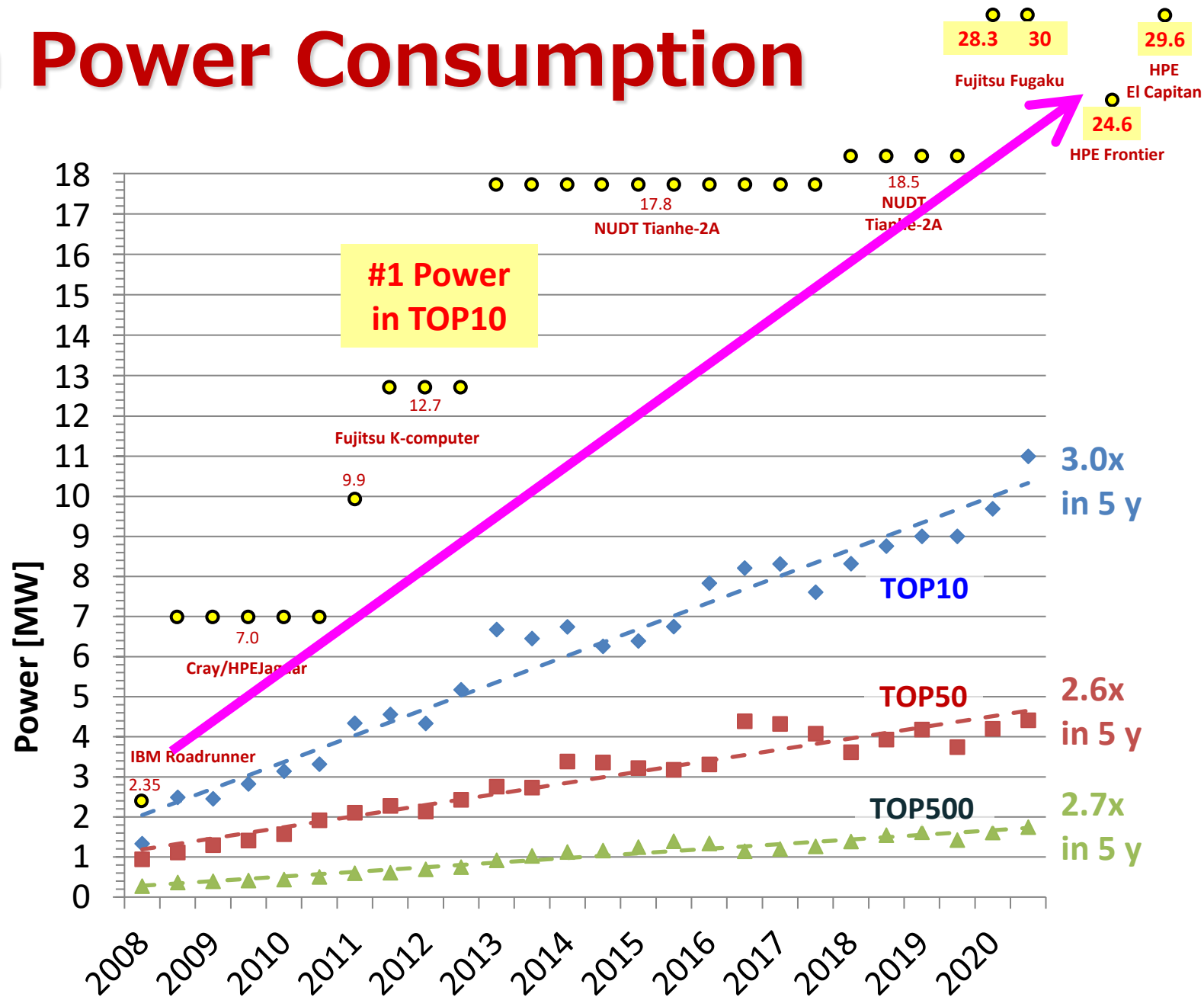
Needed to increase for higher system performance

- ✓ Limited improvement of performance per power

10s of MW for #1 HPL machines

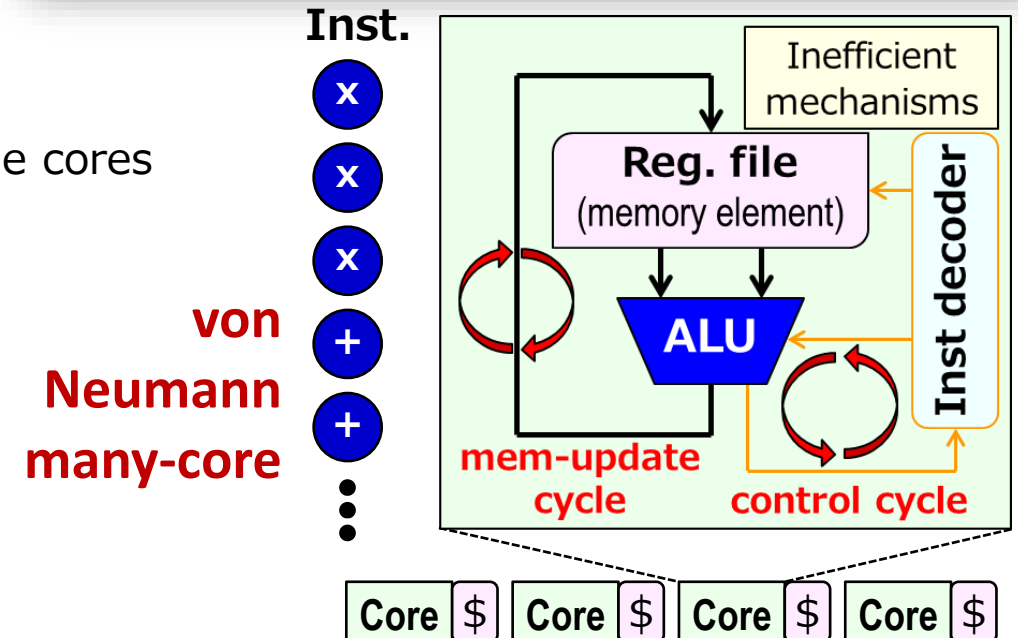
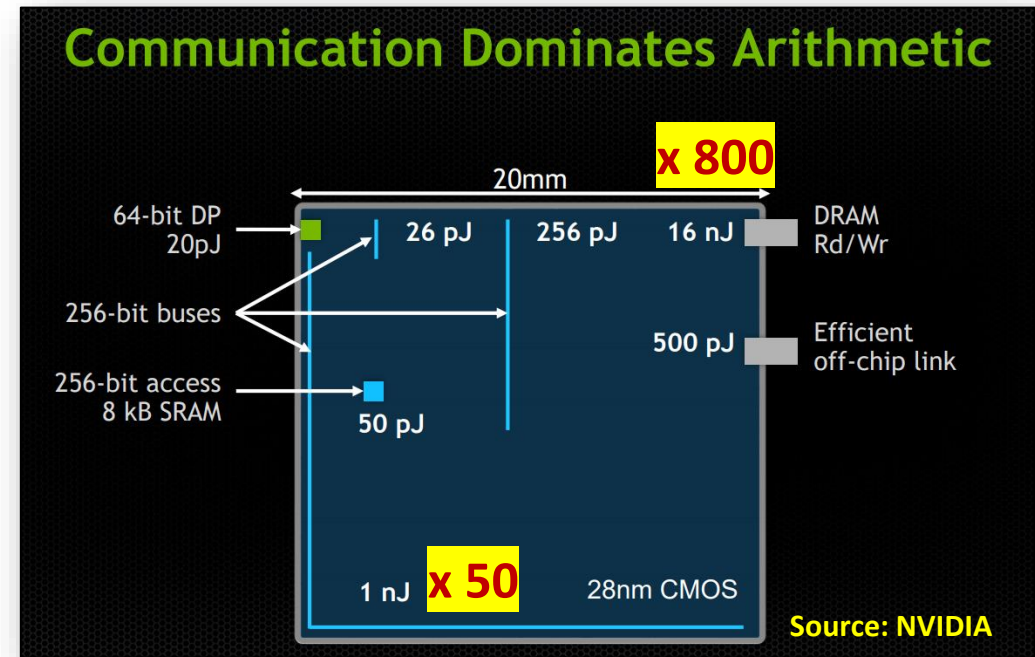
- ✓ **Fugaku 30 MW** for 442 PF
- ✓ **Frontier 24.6MW** for 1353 PF
- ✓ **El Capitan 29.6MW** for 1742 PF

System power budget
= **Critical constraint** of system performance



What Eats Power?

- **Data movement** rather than computing
 - ✓ We should remove unnecessary data movement, and make it shorter.
- **Unsuitable architecture** resulting in low efficiency and scalability
 - ✓ von-Neumann architectures (CPU & GPU) cannot efficiently scale due to
 - **memory-bottlenecked structure**; such as register files and LLC slices distributed over NoC for multiple cores
 - **Extra mechanisms** consuming power just to increase IPC such as out-of-order, branch predictor, thread scheduler
- **Recent semiconductor scaling cannot save it.**
 - ✓ Power improvement per generation is limited while it can still increase transistors per area for advanced technology nodes like 4, 2, and 1.5nm ...



Custom Data-Flow Computing

- **Data-flow computing**

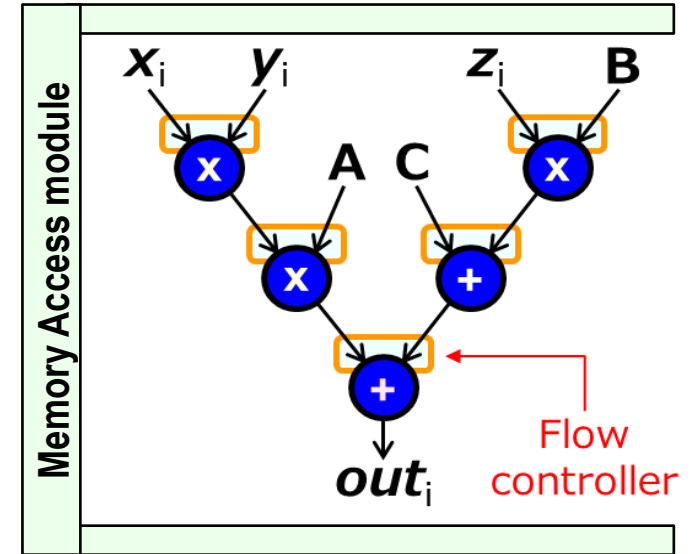
- ✓ Localized data-movement
- ✓ Lower pressure on memory access with highly pipelined computing by regular data streams
- ✓ No extra mechanisms for non-computing

- **Customization & reconfiguration**

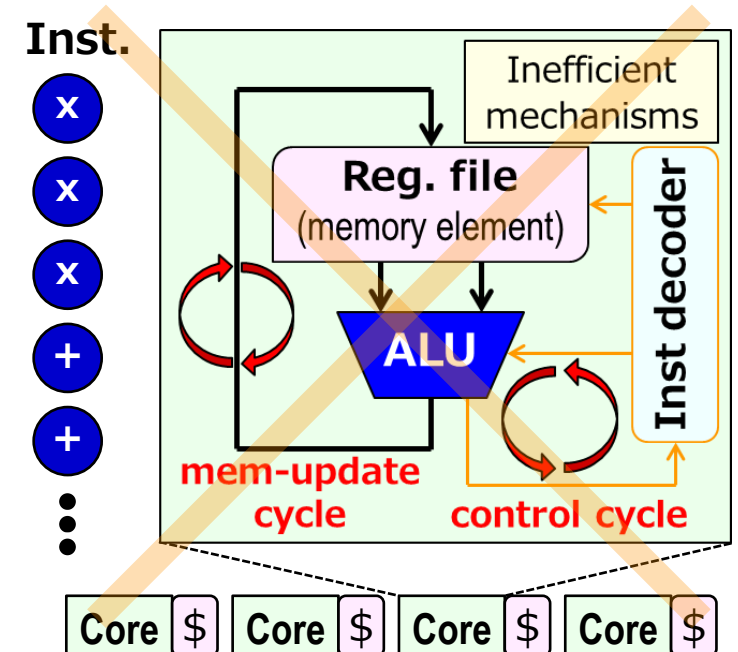
- ✓ Higher efficiency by specialization
- ✓ Programmability for various problems

What technology is suitable to for custom data-flow computing?
FPGA?

Data-flow computing



von Neumann many-core



1. Advancement of Fugaku

- ✓ Research on **Functional extension with FPGAs**
(FPGA cluster development, specialized hardware for HPC)



Experimental FPGA Cluster connected with Supercomputer Fugaku

Open-Access paper



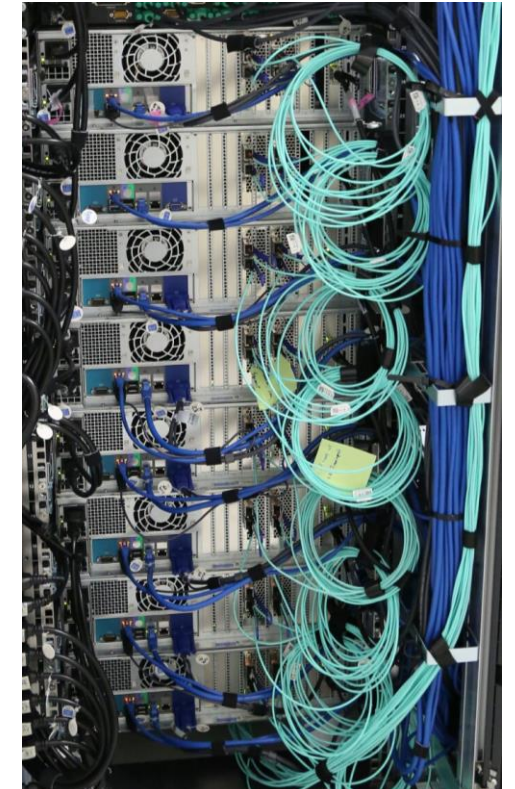
This Work



Goal : Design & demonstrate a proof-of-concept FPGA cluster for HPC research

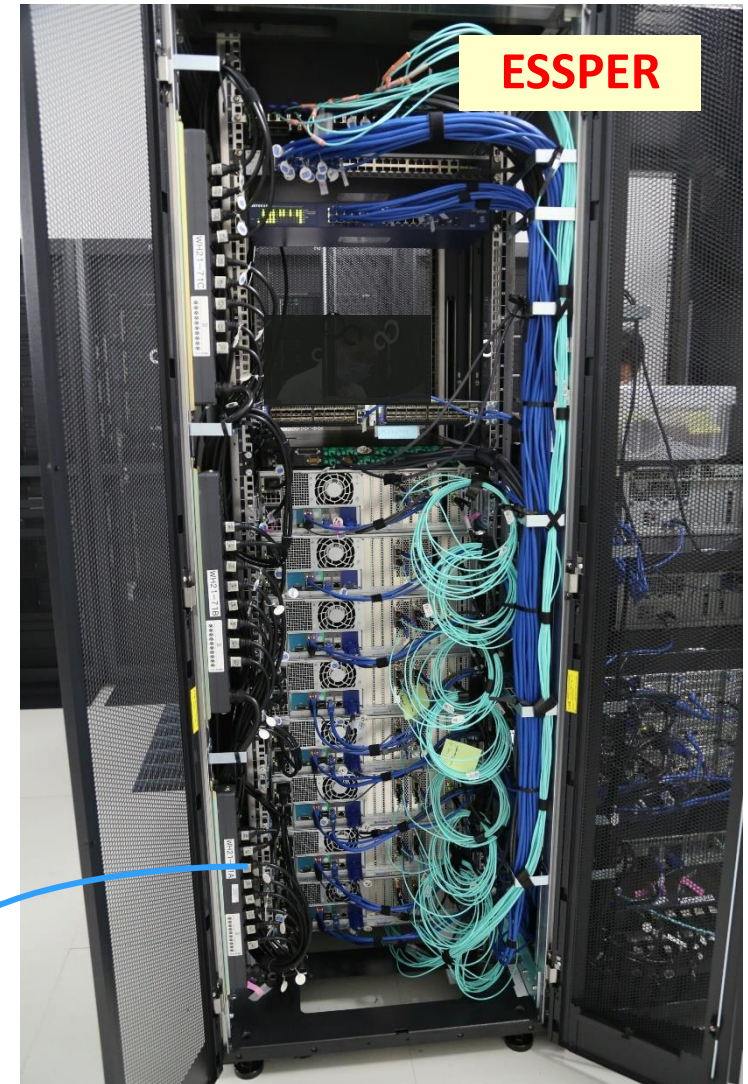
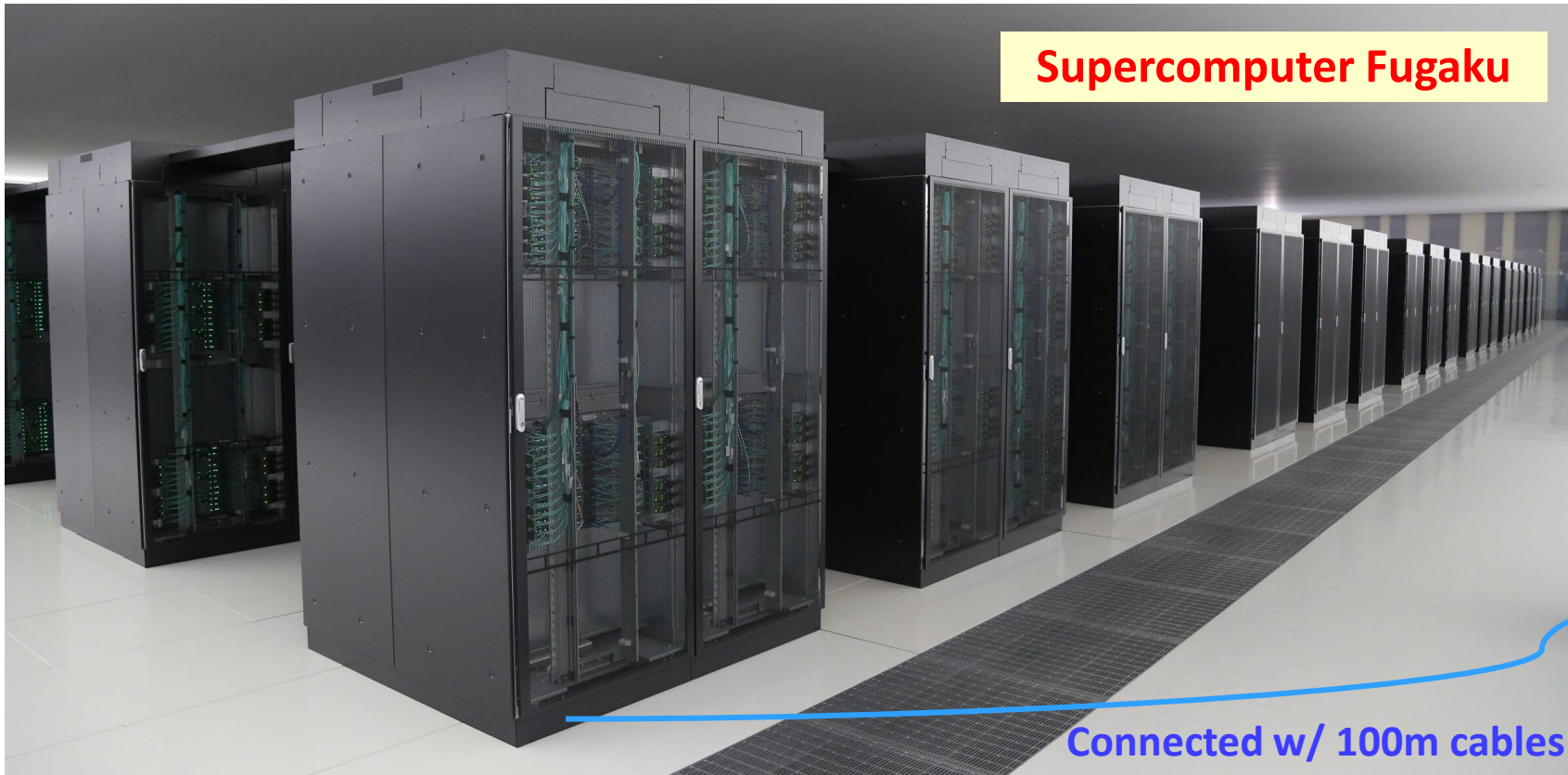
- **ESSPER** : Elastic and scalable FPGA-cluster system for high-performance reconfigurable computing
- **Contributions**
 - ✓ **Design concept** of FPGA cluster for HPC
 - ✓ **Classification** of FPGA cluster architectures
 - ✓ **Proposed system stack** with software-bridged APIs
 - ✓ **Implementation and evaluation** for FPGA-based extension of the world's top-class supercomputer, Fugaku

Open-Access paper

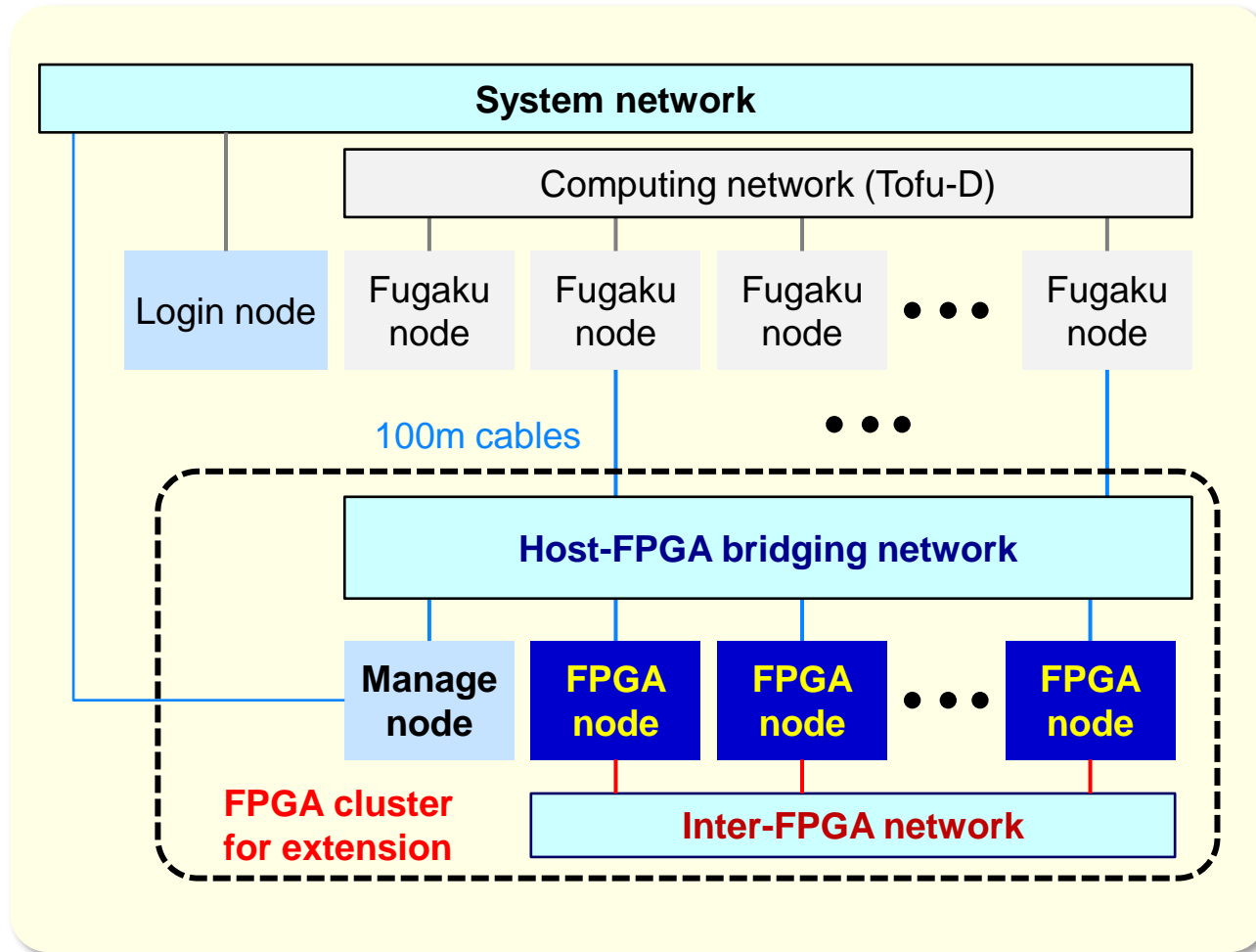


Elastic and Scalable System for High-Performance Reconfigurable Computing

Experimental prototype
for research on functional extension with FPGAs



Architecture of ESSPER



✓ Productive customizability

- No OpenCL (not limit computing models)
- FPGA Shell & HLS/HDL programming, where any hardware can be easily implemented

✓ Performance scalability

- FPGA Shell supporting high-bandwidth and low-latency network dedicated to FPGAs

✓ Interoperability

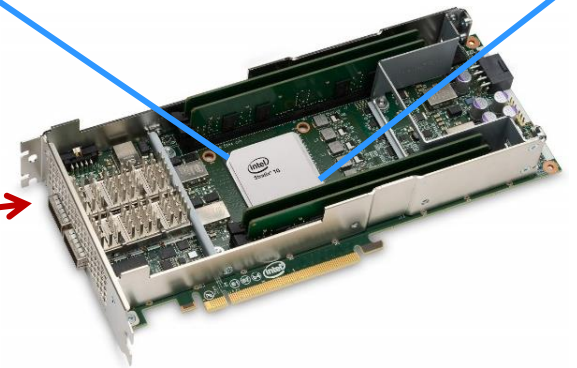
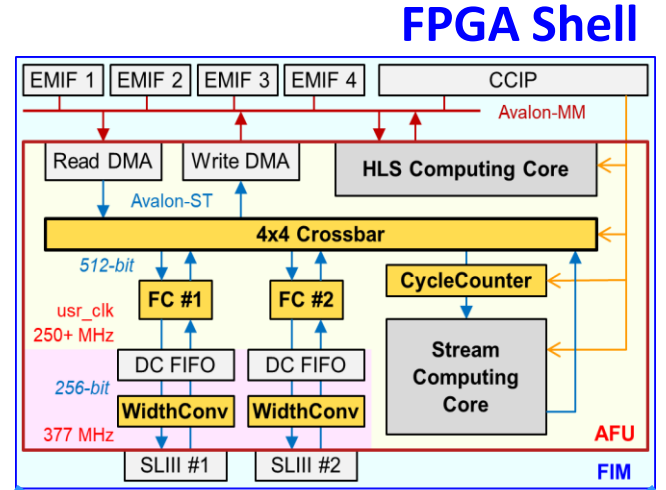
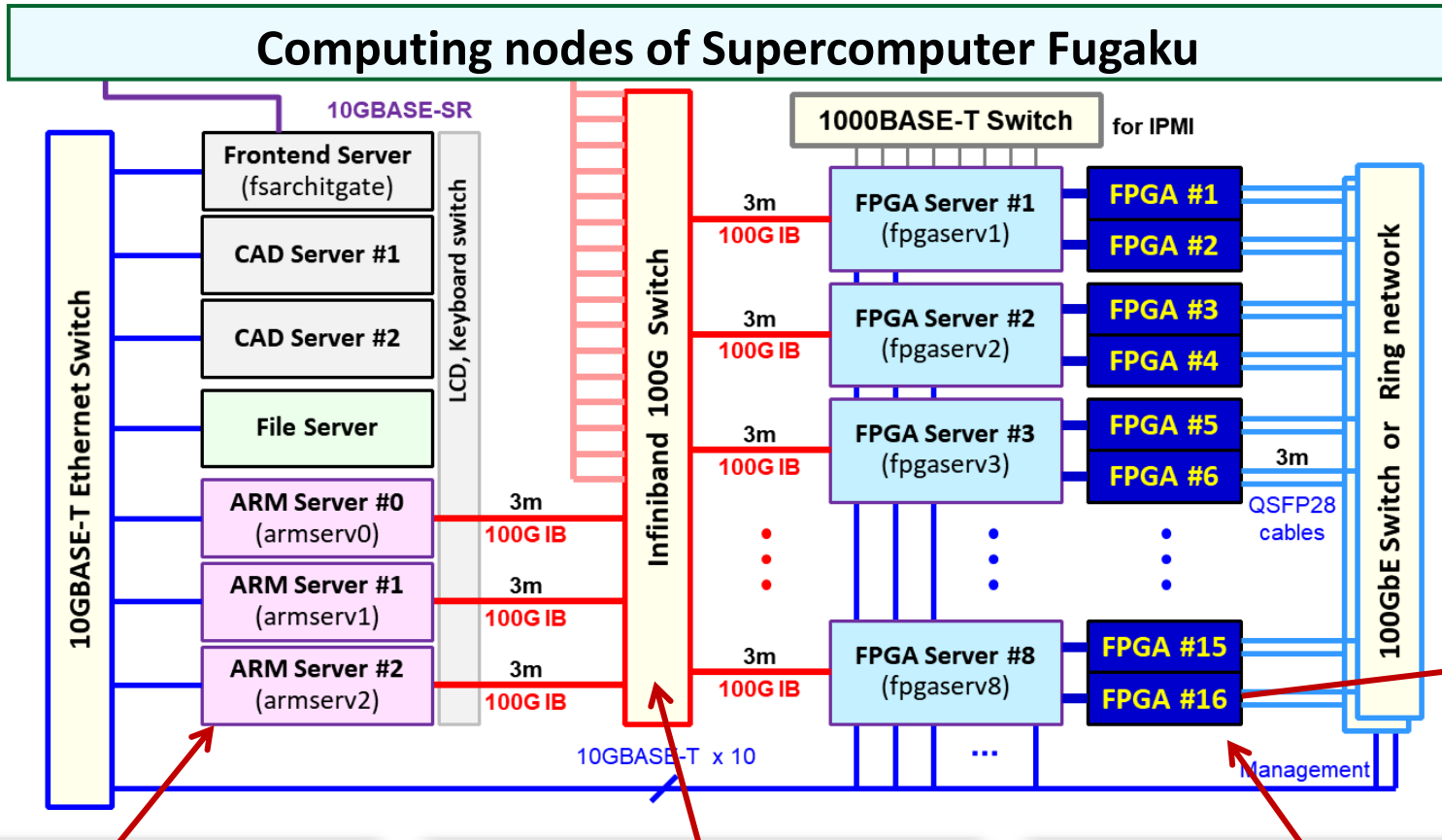
- Software-bridged driver and APIs to access FPGAs remotely through host-FPGA bridging network

ESSPER

Elastic and Scalable System
for High-Performance Re-
configurable Computing

System Design

Hardware Organization of ESSPER



Service servers

- CAD servers
- Storage server
- ARM servers

CPU - FPGA network

- 100G Infiniband
- Software-bridged driver (R-OPAE)

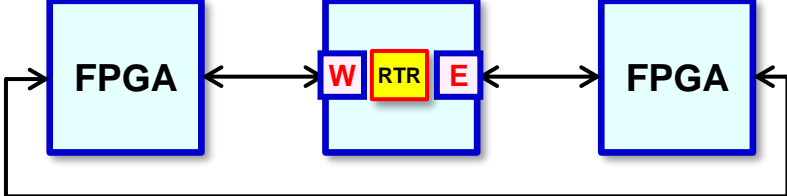
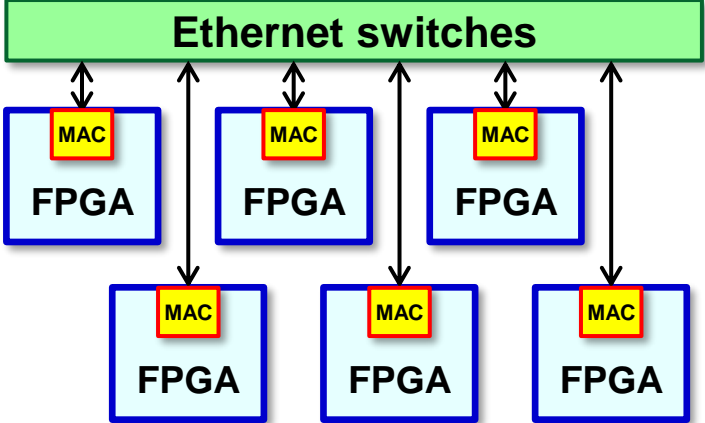
FPGA cluster

- x86 host servers
- FPGA boards
- Inter-FPGA network

FPGA Shell (SoC)

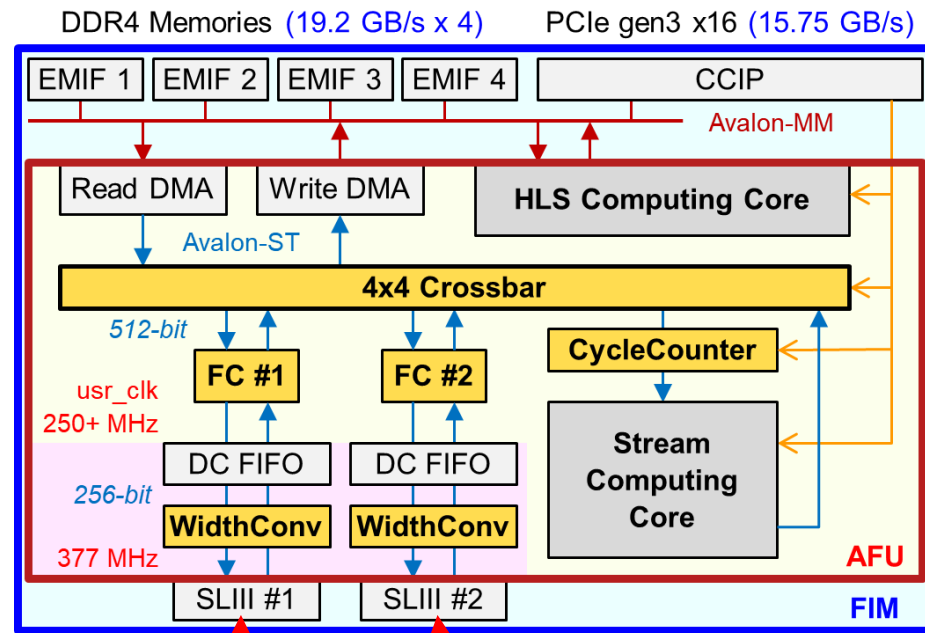
- AFU Shell design
- User HW modules can be embedded for custom computing.

Two Types of Networks

	Direct network	Indirect network
		
Characteristics	p2p-connection without switches, typical: torus network	connection with switches, typical: Ethernet
Switching	circuit or packet (w/ on-chip router)	packet
Pros	low latency, easy to use with simple HW	flexibility, small diameter, easy adoption of cutting-edge
Cons	large diameter, inflexibility in resource allocation	higher latency due to packet processing, complex and difficult to use

FPGA Shells for Direct and Indirect Networks

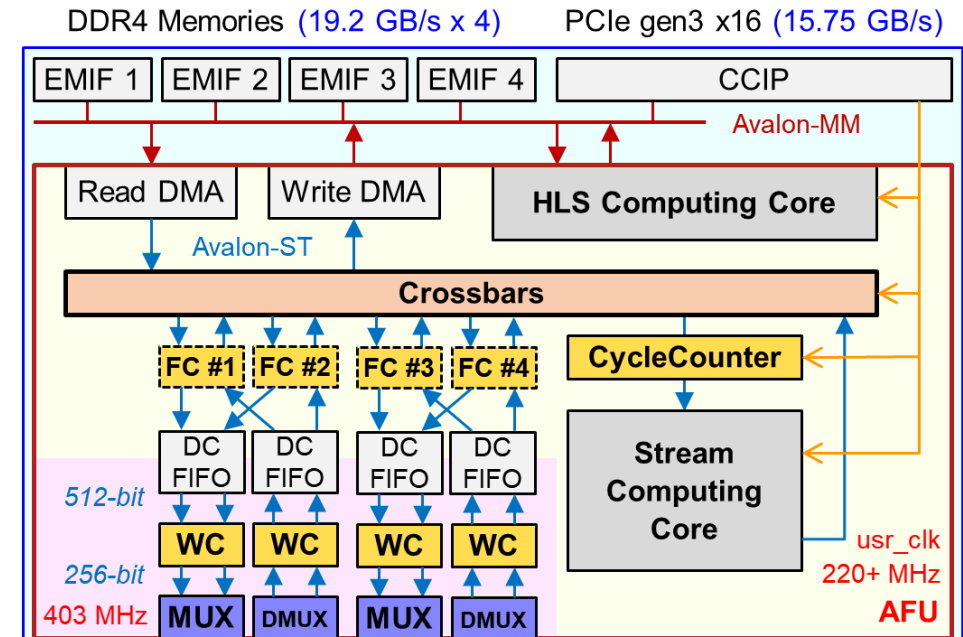
Direct connection network (DCN)



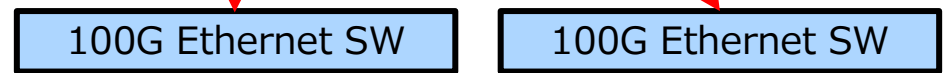
100G SerialLite3 with flow controller (FC)



Indirect network (VCSN)



100G Ethernet virtual circuit-switching nw



Open-Access paper



ESSPER

Elastic and Scalable System
for High-Performance Re-
configurable Computing

Applications, Joint Research Projects

Projects with ESSPER (Selected)

On-going (Joint) Research Projects

Hardware

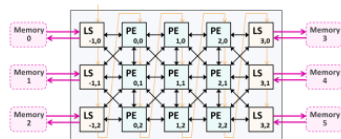
- Processor Team **CGRA**
- Kumamoto Univ **AI Engine (ReNA)**

System Software

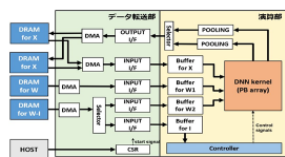
- RIKEN RPC for FPGAs
- Tohoku Univ neoSYCL (on Fugaku)

Applications

- Univ of Tokyo Bayesian network analysis
- Meiji Univ 3D FFT (presented later)
- Processor Team Fluid simulation
- Nagasaki Univ Convex method
- Hiroshima City U Breadth First Search of Graph
- Processor Team Hardwired MNIST
- JAIST Sound rendering



Riken CGRA (coarse-grained reconfigurable array)

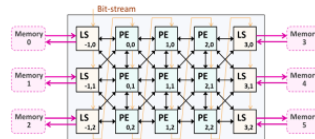


AI Engine, ReNA

Design Space Exploration of CGRA (Riken)

FPGA emulator/overlay of coarse-grained reconfigurable array (CGRA) for HPC

- Processor Research Team, Riken R-CCS
- Exploring design space of CGRA for ASIC
 - Various configurations available with library modules such as FIFO, Mux, ALU
- CGRA compiler (by Tokyo university)
 - Data-flow graph (DFG) of a loop kernel in OpenMP
 - Place and route by Genetic Algorithm
 - Benchmarking (Stencil, Convolution, FFT, etc.)
- Initial design completed
 - System Verilog
 - Verified by RTL simulation
 - Preparing for FPGA-based implementation



Overall structure of CGRA (size: parameterized)

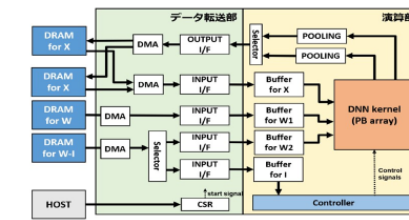


Mapping examples on CGRAs (16x16, 8x16)

ReNA: Architecture for CNN Inference (Kumamoto U)

Transplant Inference processor ReNA developed for edge ASIC to FPGA

- Laboratory of Prof. Iida @ Kumamoto U
- Achieve highly-scalable inference with multiple FPGAs
 - Extend the processor over FPGAs using inter-FPGA network
- 64x64 systolic array
 - FMA x 64² = 8192 parallel
 - Convolution and all-to-all computations optimized by dedicated mappings
 - Various models available
- Initial implementation completed for single FPGA
 - Verilog HDL

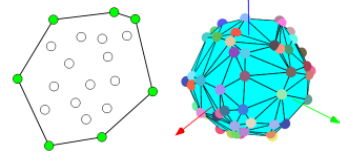


Architecture of ReNA

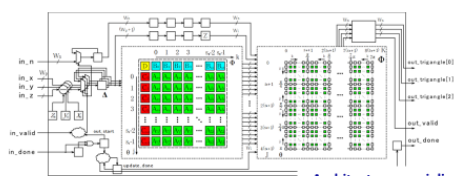
Architecture for Convex Hull Generation (Nagasaki U)

Acceleration of Convex Hull Generation with point clouds using multiple FPGAs

- Laboratory of Prof. Shibata @ Nagasaki U
- Applications of Convex Hull
 - Delaunay diagram construction/area estimation/registration/image processing, etc.
 - Object collision detection
 - Approximation of moving objects and obstacles in path planning
 - physics simulator
 - Real-time rendering of point clouds
- Pipelining for higher throughput and lower latency than GPUs
- Initial implementation completed
 - SystemVerilog
 - Comparison with Qhull software



Convex hull : The smallest convex set enclosing a point set

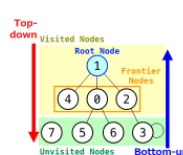


Architecture specialized for convex hull generation

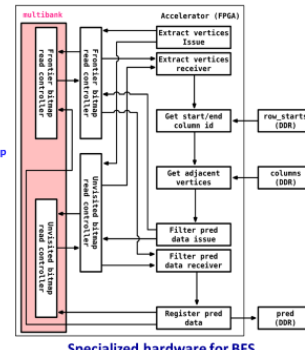
Specialized Hardware for BFS by HLS (Hiroshima city U)

BFS Accelerator HyGTA

- Laboratory of Prof. Tanigawa @ Hiroshima city U
- Hybrid Graph Traversal Accelerator
 - Hybrid algorithm combining Top-down and Bottom-up searches
- Implement by HLS, demonstrate and evaluate with FPGA
- Pipelining, latency hiding, efficient memory sub-system
 - Pipelined BFS
 - Cache memory for adjacent-node data
 - Multi-banked bitmaps for visited-node record
 - Effective use of memory access patterns specific in BFS



Graph500 Ranking (BFS as of Nov, 2021)			
RANK	MACHINE	SCALE	GTEPS
32	ENIAD (FPGA)	26	783.75

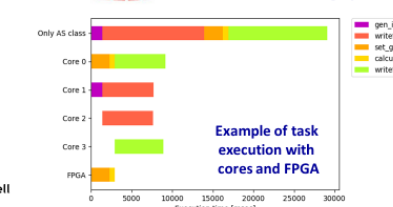
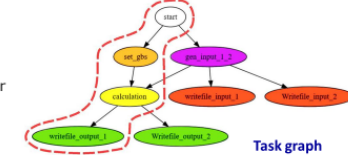
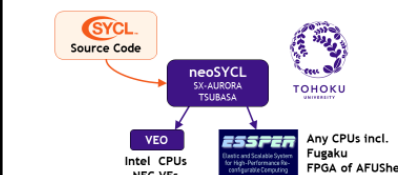


Specialized hardware for BFS accelerator "HyGTA"

Task off-loading to FPGAs by own SYCL (Tohoku U)

neoSYCL : yet another SYCL implementation

- Laboratory of Prof. Takizawa @ Tohoku U
- neoSYCL originally developed for NEC Vector Processor
- Support FPGA and AFUShell of ESSPER
- Dynamic task scheduler
- Tasks can be off-loaded from Fugaku to FPGAs.

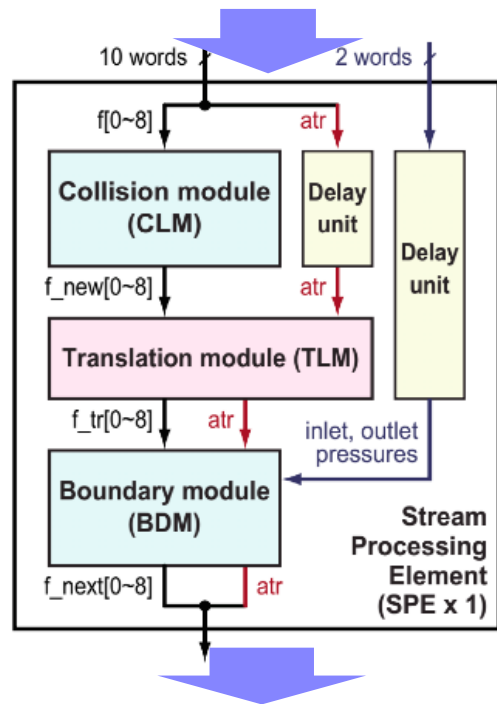
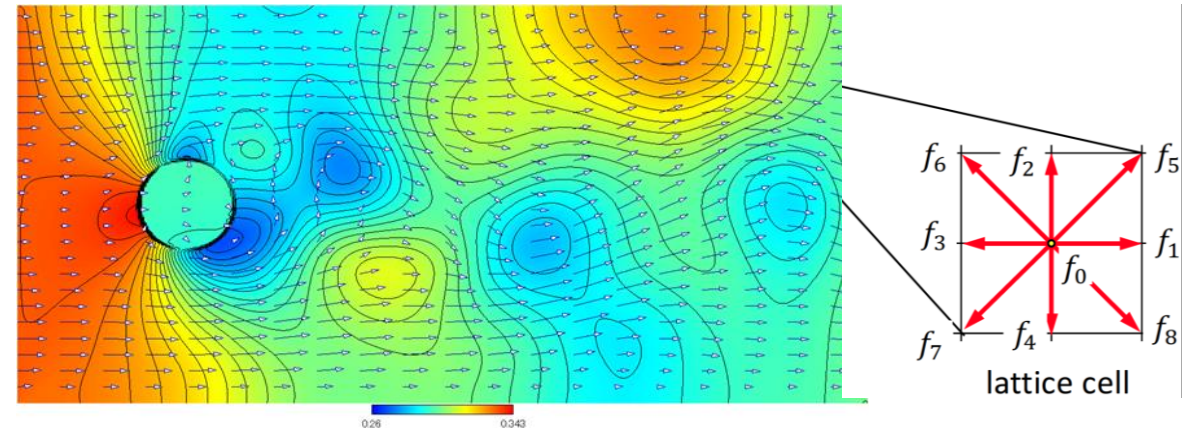


Example of task execution with cores and FPGA

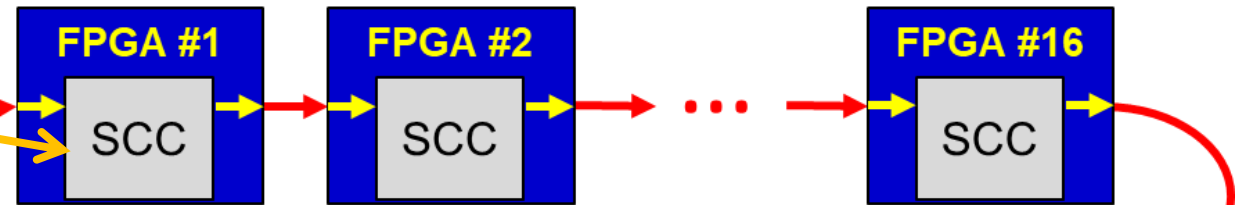
Multi-FPGA Application using Direct Connection Network

- **Stream computing of Fluid simulation with multiple FPGAs**

- ✓ Lattice Boltzmann method (LBM)
- ✓ Extended pipeline with ringed FPGAs



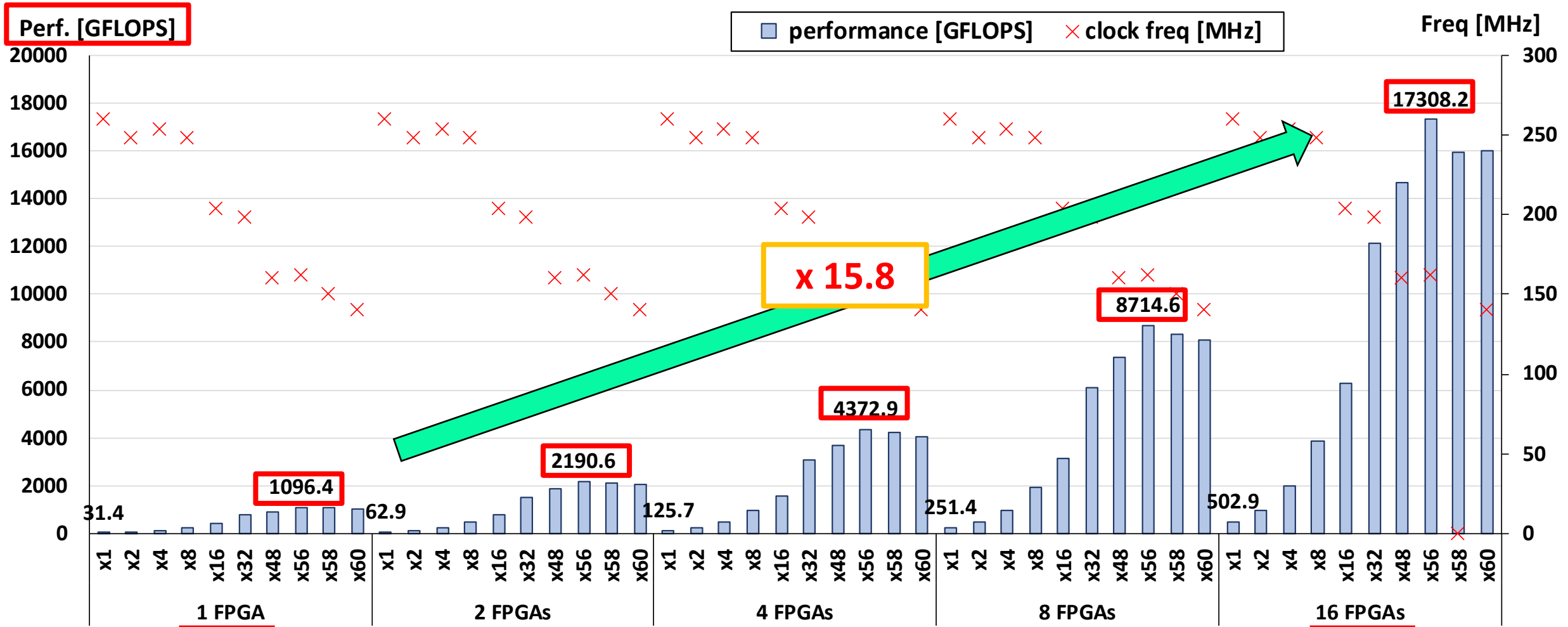
Stream
computing
core (SCC)



FPGAs in a ring

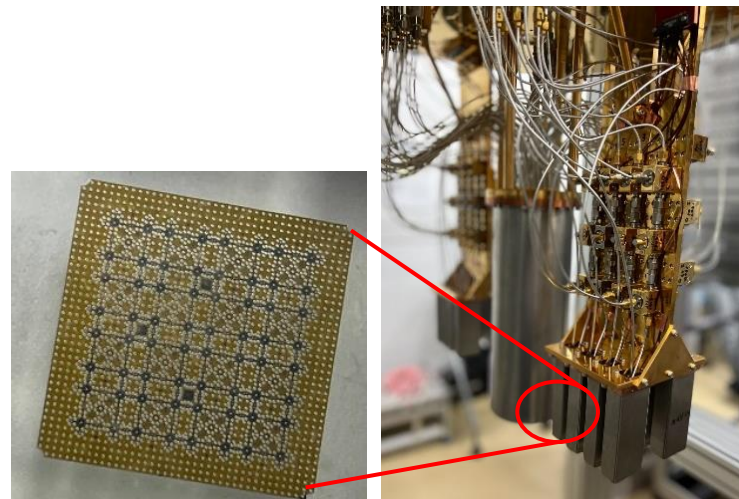
Performance of 2D LBM with 100Gbps Ring NW

Computational performance (FLOPS) when processing about 2GB data

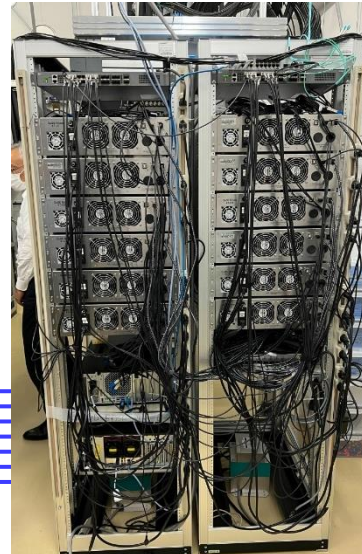


Quantum Error Correction with FPGAs

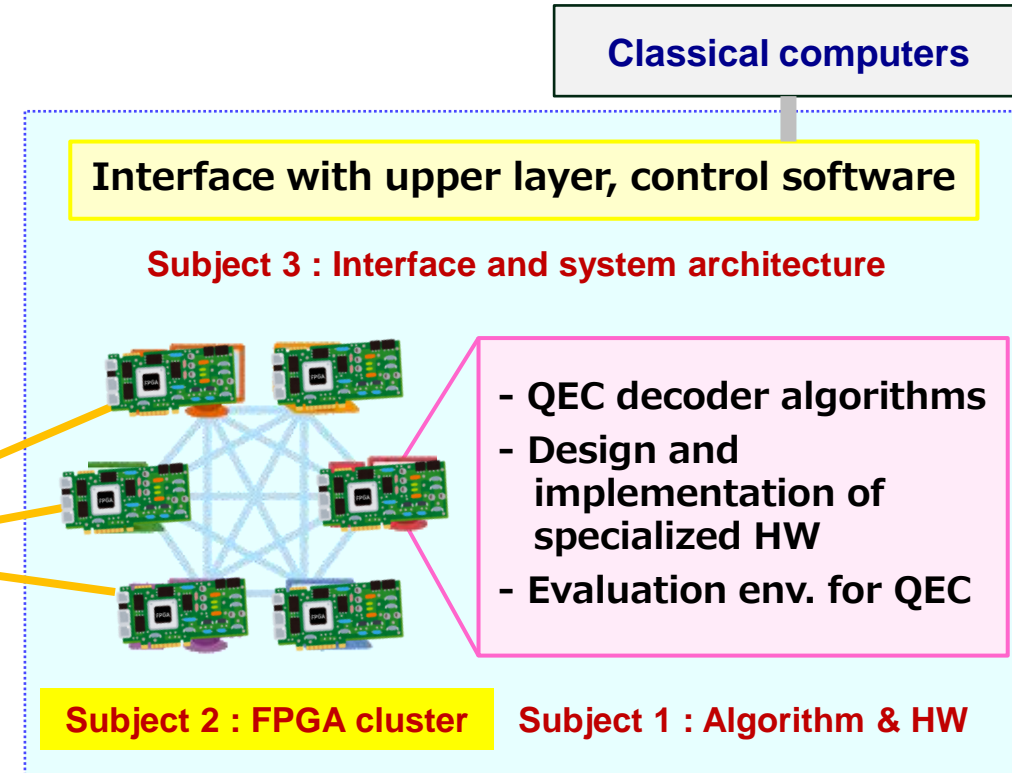
- **Fault-tolerant quantum computers (FTQC) using quantum error correction (QEC)**
 - ✓ Need to solve **minimum-weight perfect matching (MWPM)** problem
 - ✓ Need to encode **1000 logical qubits using 1M physical qubits** finally
 - ✓ Scalability and **low-latency (< 10us)** are required.
- **Goal**
 - ✓ Explore scalable QEC hardware algorithm and system
 - ✓ Demonstrate for proof-of-concept



RIKEN's superconducting qubits

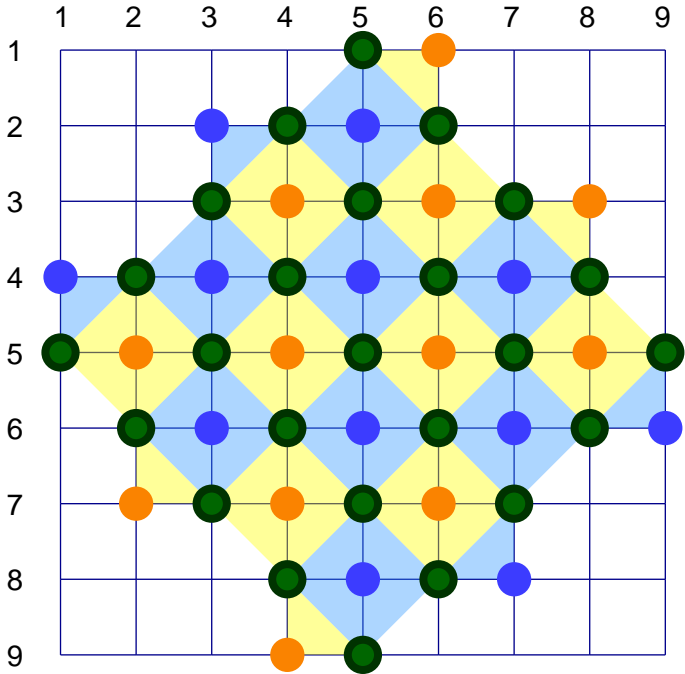


Quantum-Classical Frontend



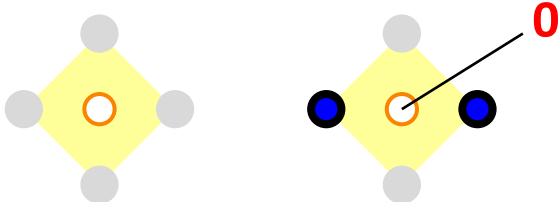
Backend system for QEC

Surface Code for Quantum Error Correction

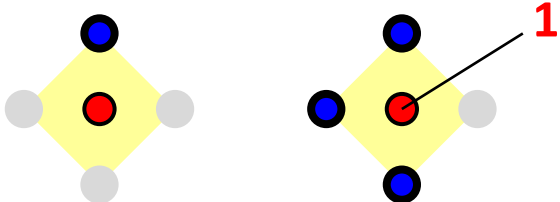


Surface code with code distance of $d = 5$
(single logical qubit)

- Syndrome = 0 for even errors

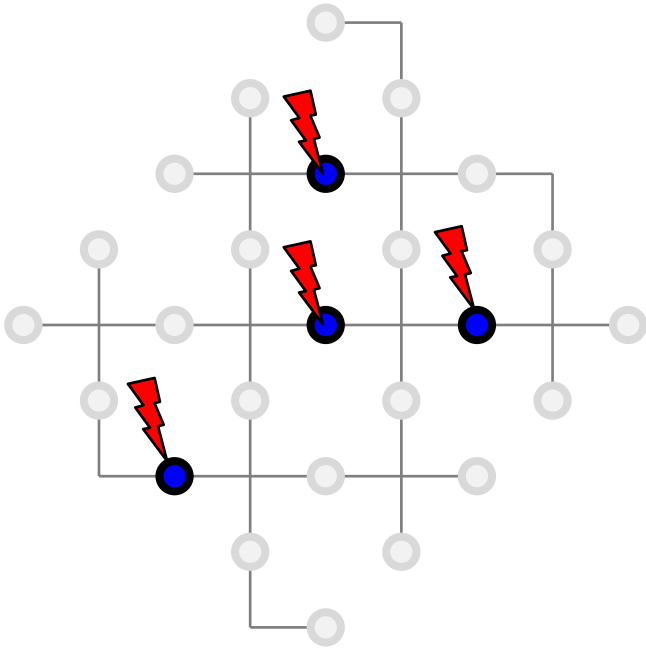


- Syndrome = 1 for odd errors

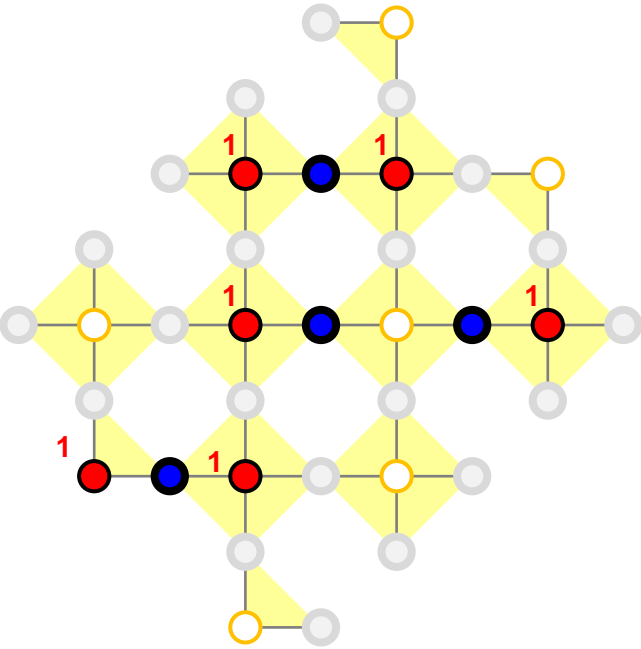
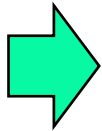


Parity measurement for data errors
(Same for X and Z, respectively)

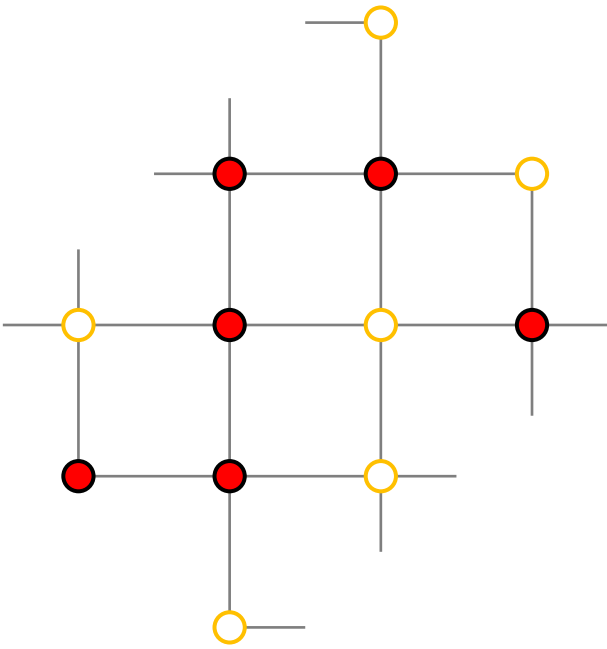
Surface Code for Quantum Error Correction



Data qubits with errors



Syndrome Measurement

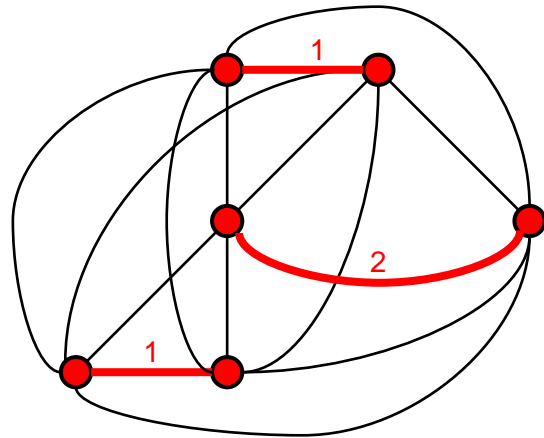


Decoding graph

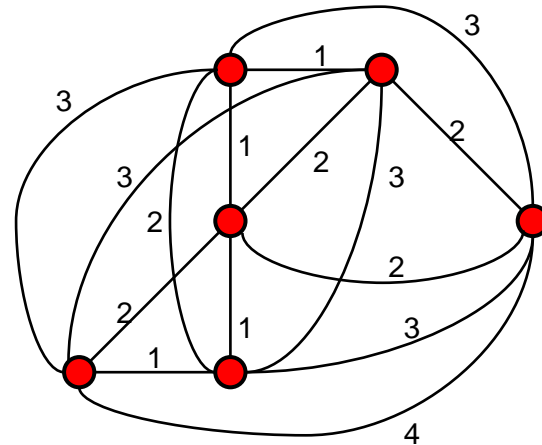
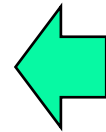
(What we know as syndrome)

How can we know where the errors exist ?

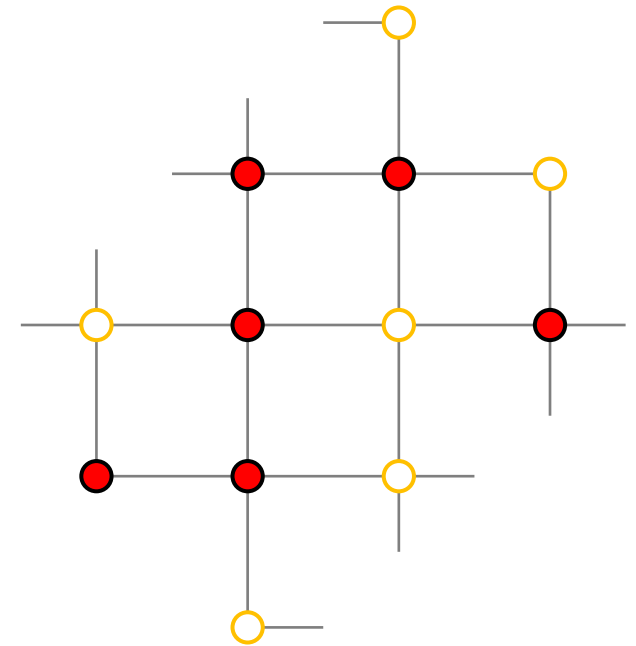
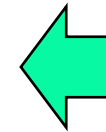
Minimum-Weight Perfect Matching in Syndrome Graph



**Minimum-weight
perfect matching (MWPM)**



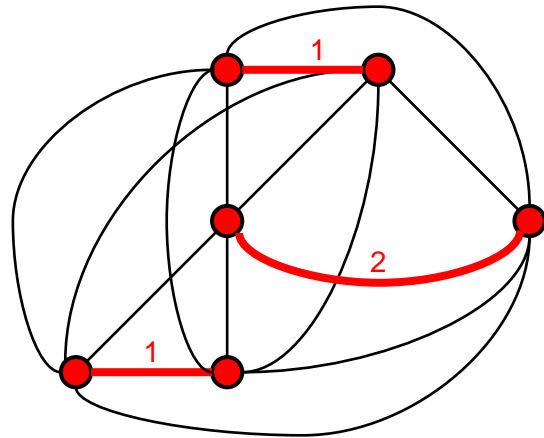
**Syndrome graph
with weight per edge
(weight \sim Manhattan dist.)**



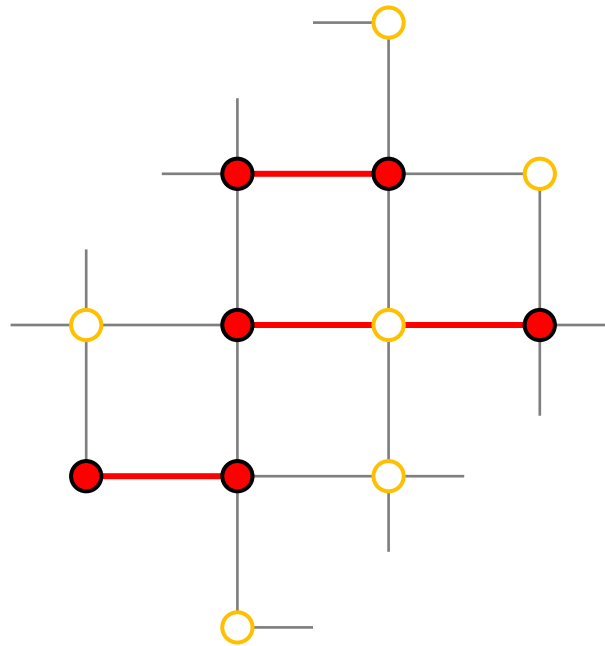
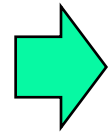
**Decoding graph
(What we know as syndrome)**

Decoding Results for Most likely Errors

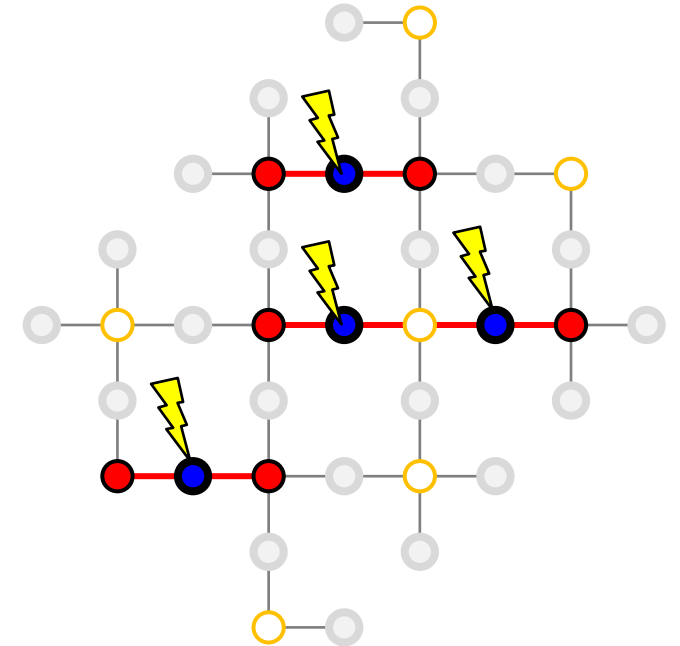
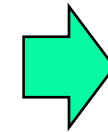
Need to handle 3D lattice for measurement errors.



Minimum-weight perfect matching (MWPM)

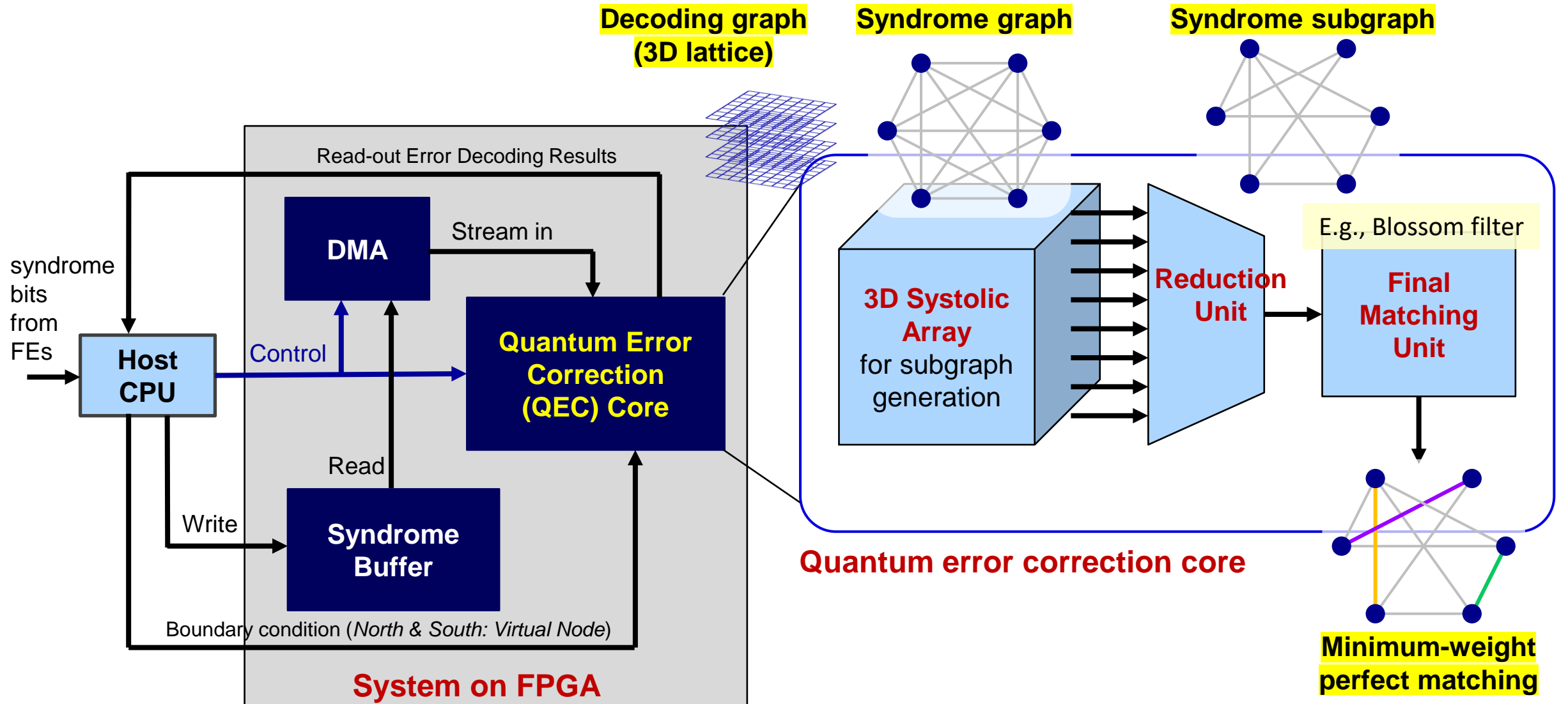


MWPM paths in Decoding graph



Most likely errors of data qubits in the paths

Hardware Design for Syndrome Subgraph Decoder



Lessons Learned with **ESSPER**

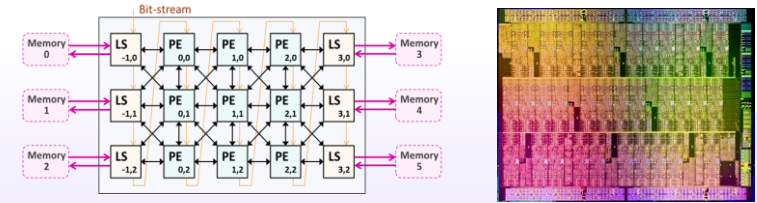
Open-Access
paper



- **FPGA-based reconfigurable computing works.**
- **Productivity is not high, especially for multiple FPGAs.**
 - ✓ Even HLS requires know-how on optimization.
 - ✓ Lack of debugging tool, and simulation environment.
- **High scalability, but FP performance is lower than GPUs for major domain.**
 - ✓ FPGA has **higher overhead (area, power, freq)** and lower memory bandwidth.
 - Sometimes, FPGA's **resource balance doesn't fit** requirement (e.g., **insufficient on-chip RAM**).
 - ✓ **Customization** with FPGA can give better solution for some specific requirement:
 - E.g., non-numerical & low-latency for quantum error correction
- **Reconfigurable data-flow itself should be Okay, but**
How can we make it a first-class citizen in HPC?

2. Exploration of new HPC & AI architectures

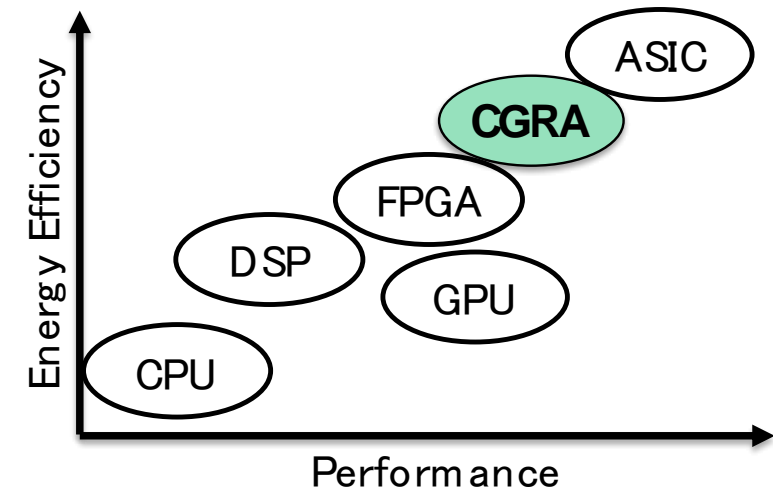
- ✓ Research on reconfigurable accelerator (e.g. **CGRA**)
- ✓ Research on next-generation **AI chip architecture**



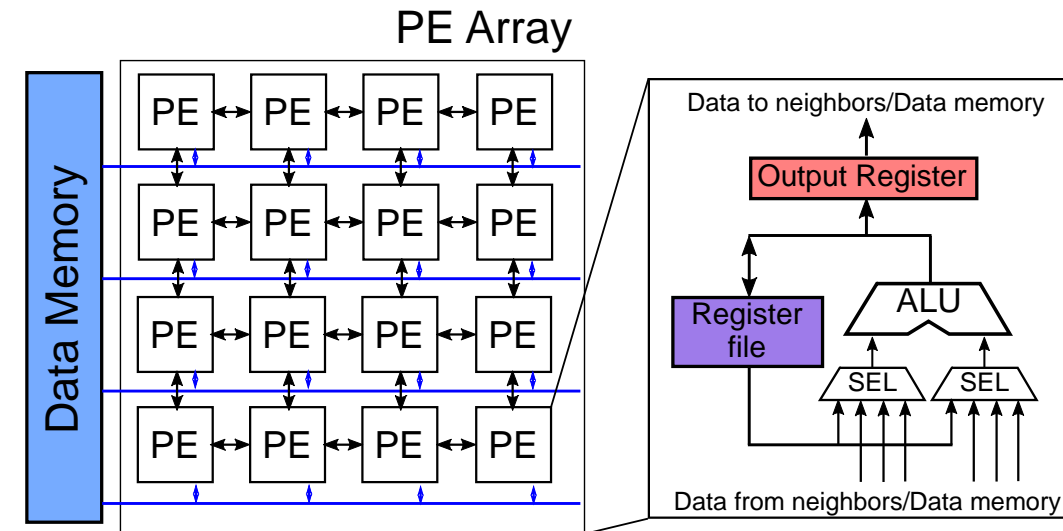
Coarse-Grained Reconfigurable Array for HPC (and AI)

Coarse-Grained Reconfigurable Array (CGRA)

- **Architecture for data-driven computing**
 - ✓ Composed of an **array of processing elements (PEs)**, where we map DFGs for computing
 - ✓ Provide a **word-level reconfigurability** (e.g., 32-bit)
 - ✓ **Higher energy efficiency than FPGAs** (of bit-level)
 - ✓ **Performance close to ASIC**-based accelerators
- **Application area of CGRAs**
 - ✓ Traditionally, targeted for lower-power embedded apps, e.g., image processing
 - ✓ Recently, expected for hi-performance deep-learning
- **Questions**
 - ✓ CGRAs also promising for HPC?
 - ✓ What architecture/design decision required HPC?



Comparison with other architectures [1]



General structure of the CGRAs [2]

[1] Liu, Leibo, et al. "A survey of coarse-grained reconfigurable architecture and design: Taxonomy, challenges, and applications." *ACM Computing Surveys (CSUR)* 52.6 (2019): 1-39.

[2] Takuya Kojima, et al., "Exploration Framework for Synthesizable CGRAs Targeting HPC: Initial Design and Evaluation," *Procs. CGRA4HPC*, May 30-June 3, 2022.

HPC Performance Requirement in Roofline Model

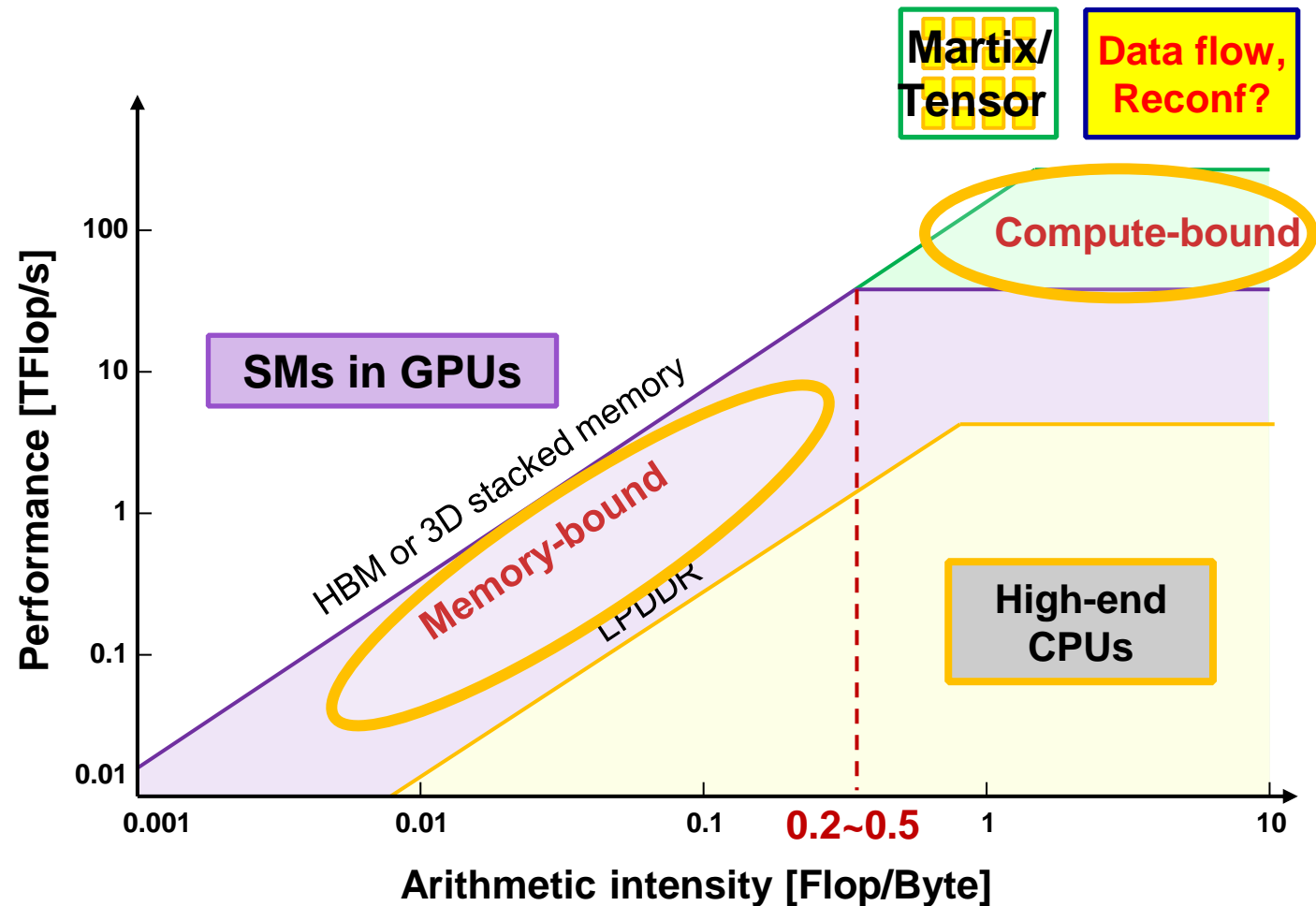
- Roofline model

- ✓ Peak performance available according to *arithmetic intensity*
- ✓ *Memory-bound or Compute-bound*

- Steaming processor can cover memory-bound applications.

- What architecture should be applied to compute-bound?

- ✓ Higher compute density
- ✓ Higher performance per power



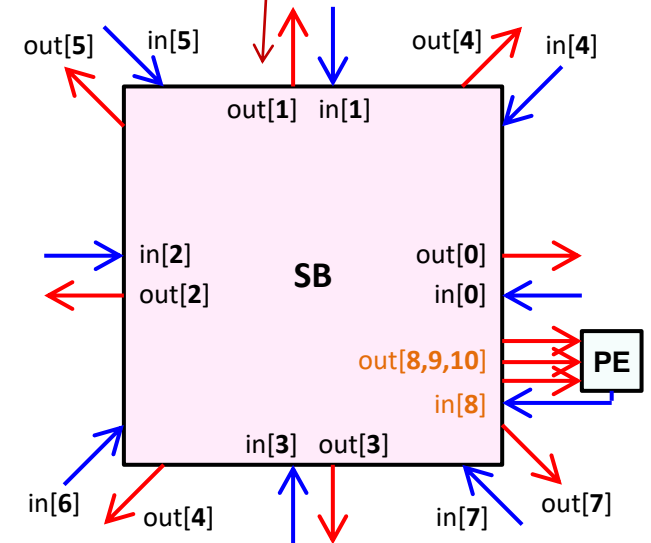
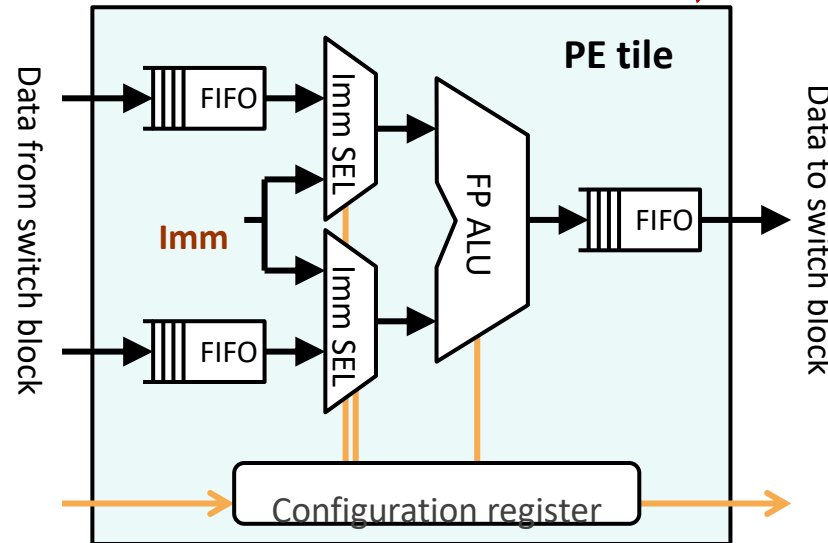
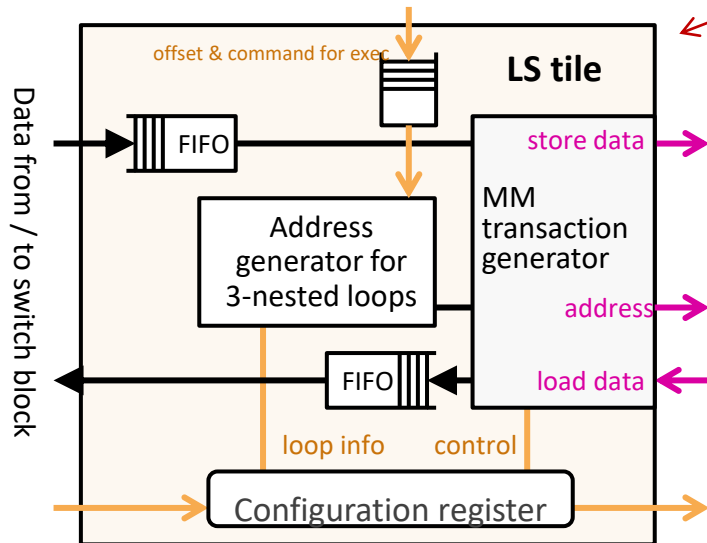
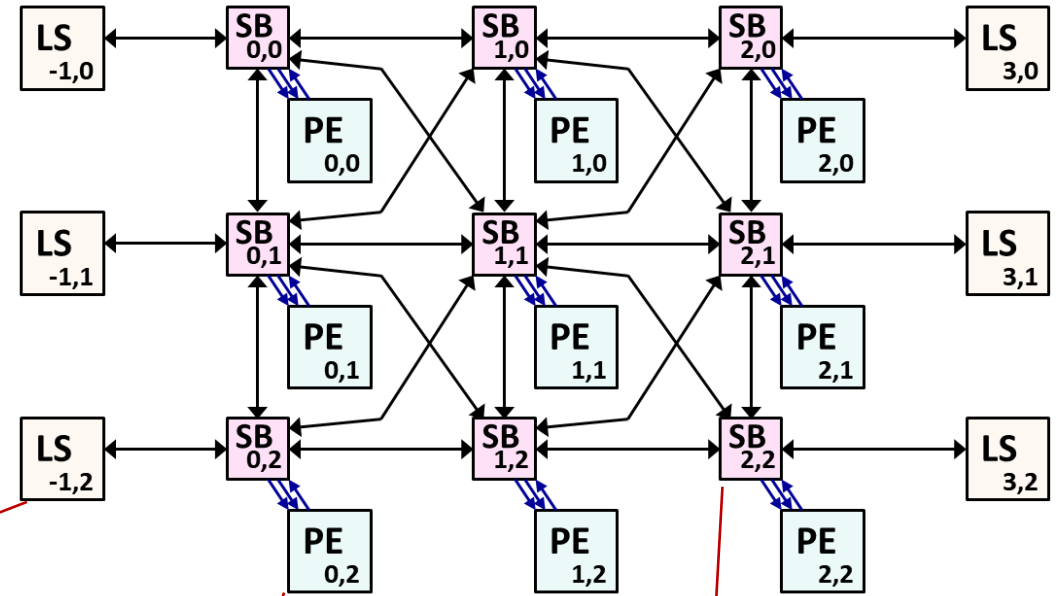
Roofline model for different performance characteristics

Exploration of Trade-offs Between General-Purpose and Specialized Processing Elements in HPC-Oriented CGRA

Emanuele Del Sozzo, **Xinyuan Wang**, Boma Adhi,
Carlos Cortes, **Jason Anderson**, Kentaro Sano
(Presented at IPDPS'24)

RIKEN Baseline CGRA

- **HPC-oriented CGRA** with the following design philosophy
 - ✓ **Modular design** for design space exploration with various configuration and sizes
 - ✓ **Isolation** between computation and memory access
 - ✓ **Floating-point** operation capability for HPC



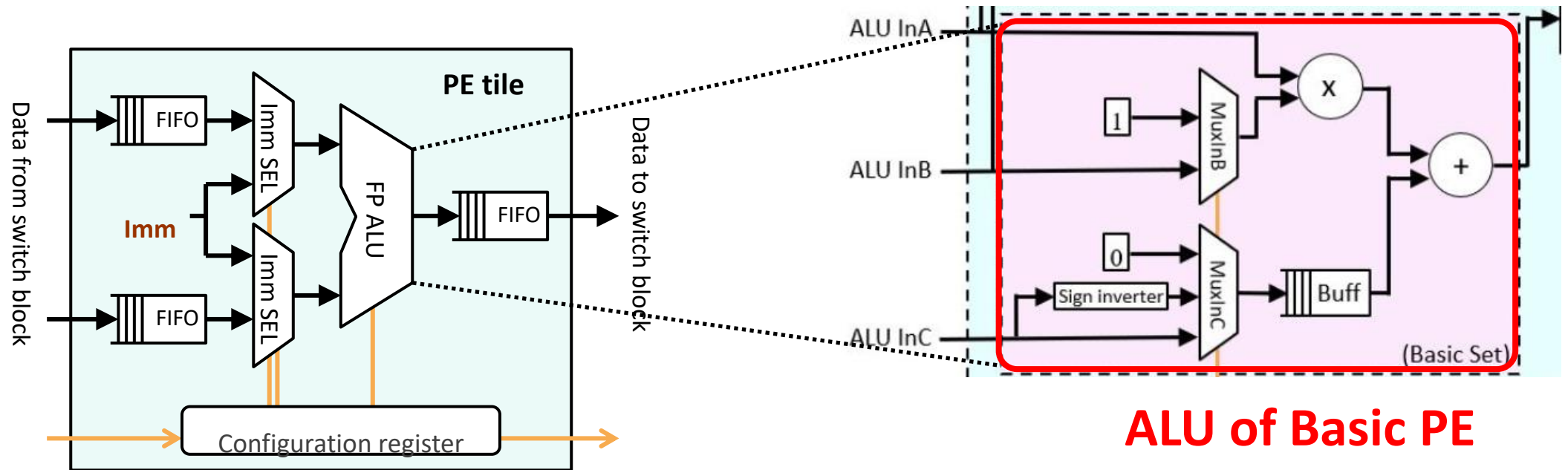
Heterogeneity for HPC

- Extend CGRA with different types of PEs

- ✓ **Basic PE :** Add, Sub, Mul, FMA

- ✓ Complex PE : Exp, Log, Sqrt, Div

- ✓ Full PE All of them

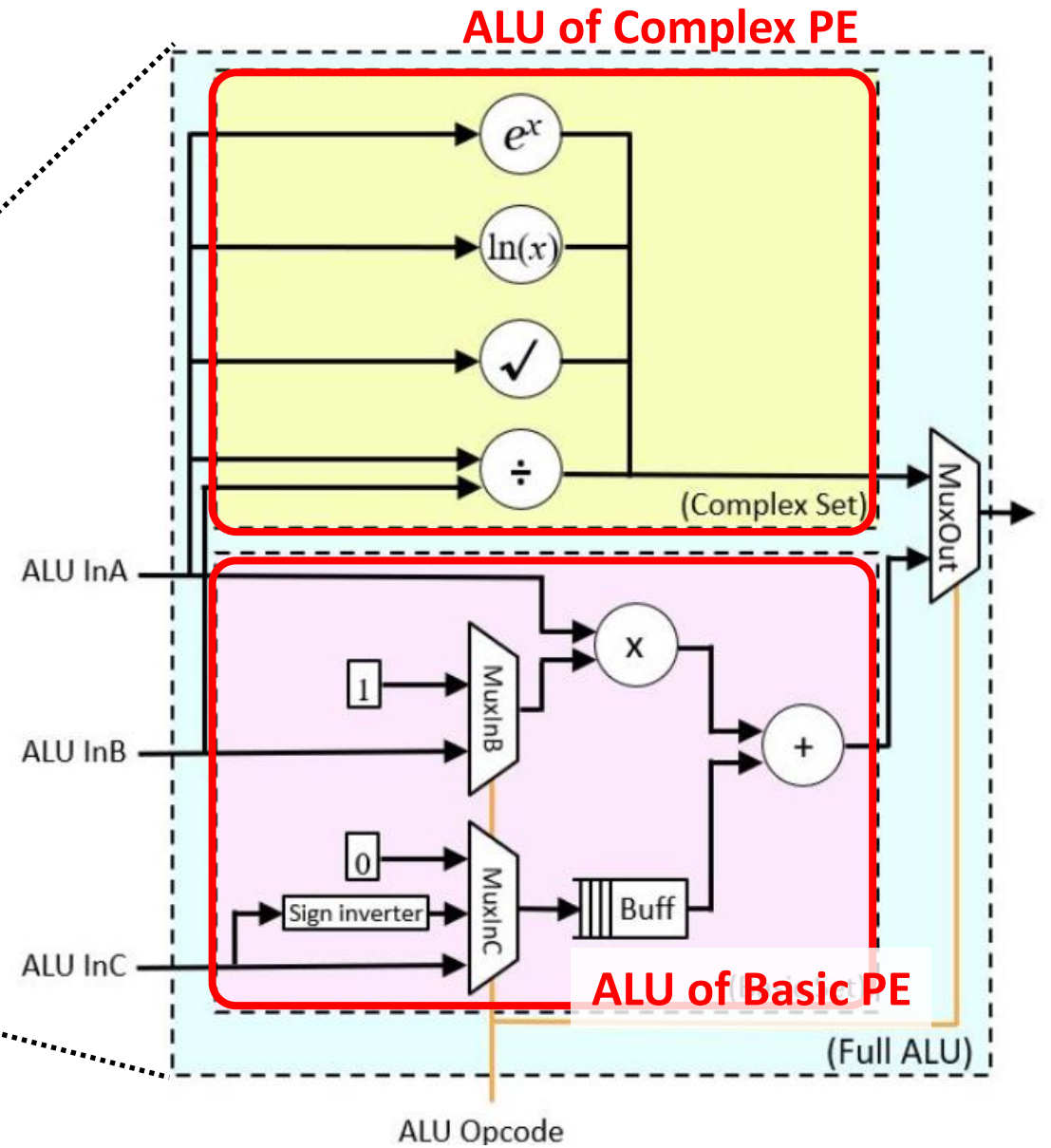
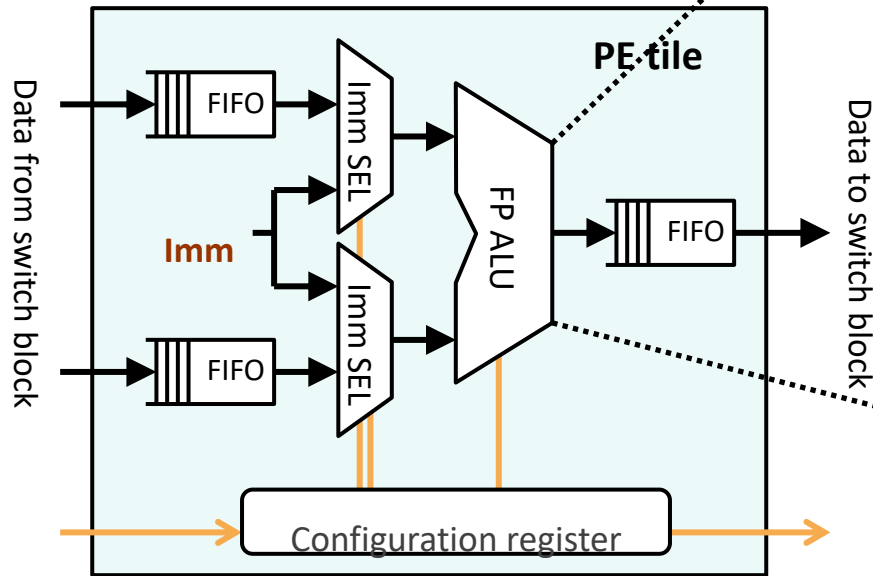


ALU of Basic PE

Heterogeneity for HPC

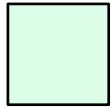
- **Extend CGRA with different types of PEs**

- ✓ Basic PE : Add, Sub, Mul, FMA
- ✓ Complex PE : Exp, Log, Sqrt, Div
- ✓ **Full PE** **All of them**

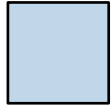


ALU of Full PE

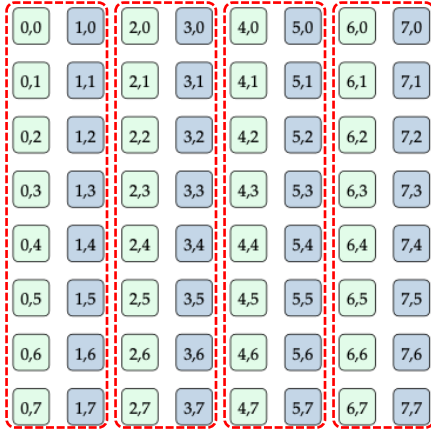
Chip Floorplans of Heterogeneous CGRA



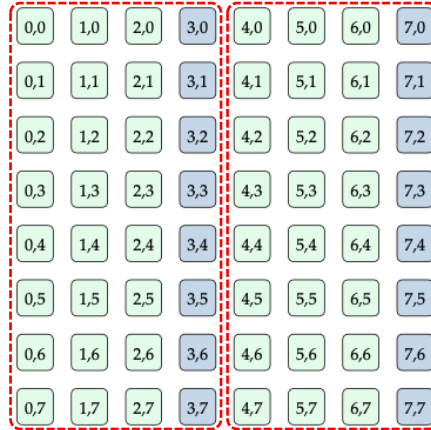
BASIC PE



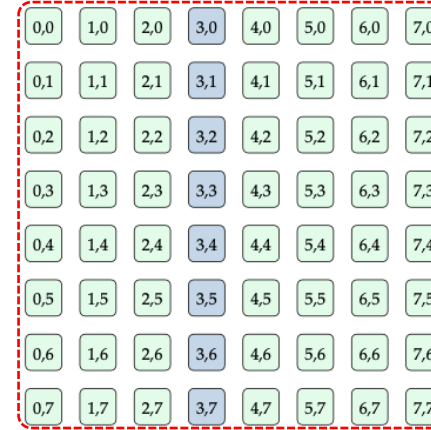
**COMPLEX/
FULL PE**



1:1 column floorplan



3:1 column floorplan



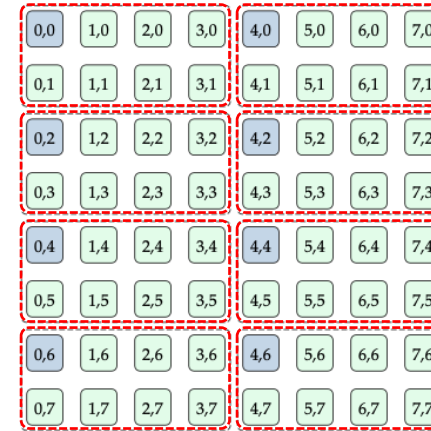
7:1 column floorplan



1:1 cluster floorplan



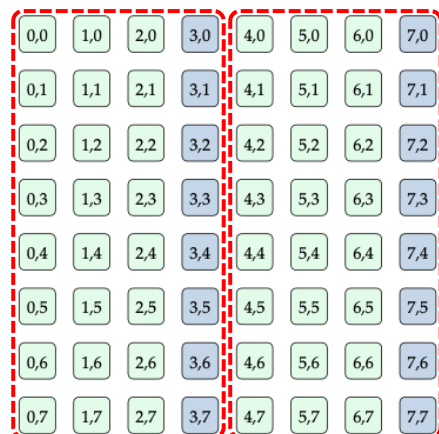
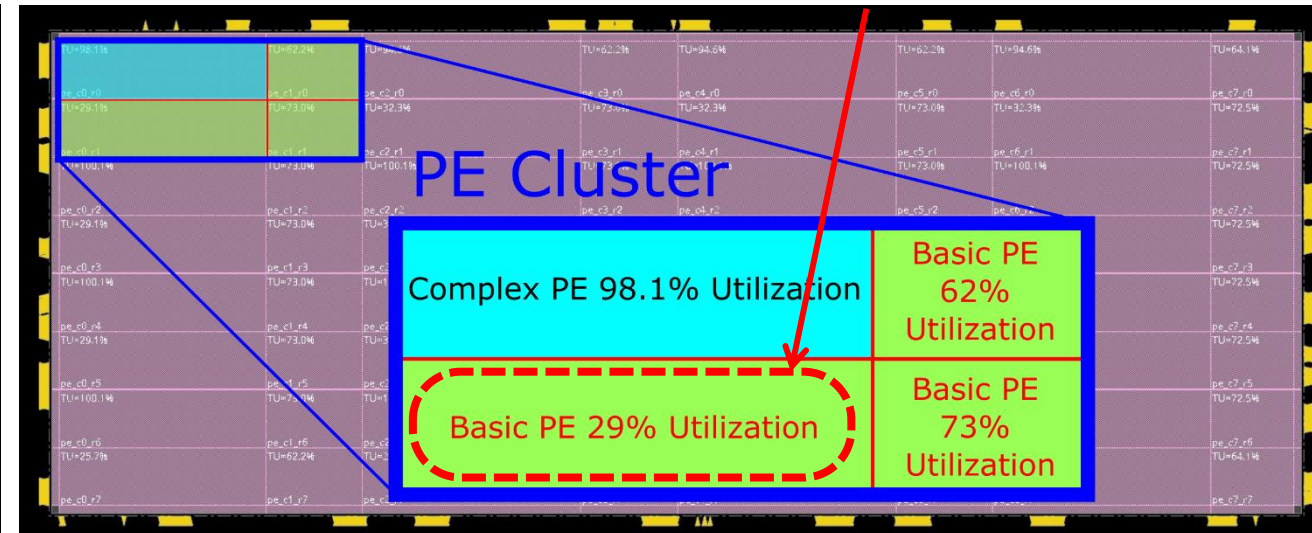
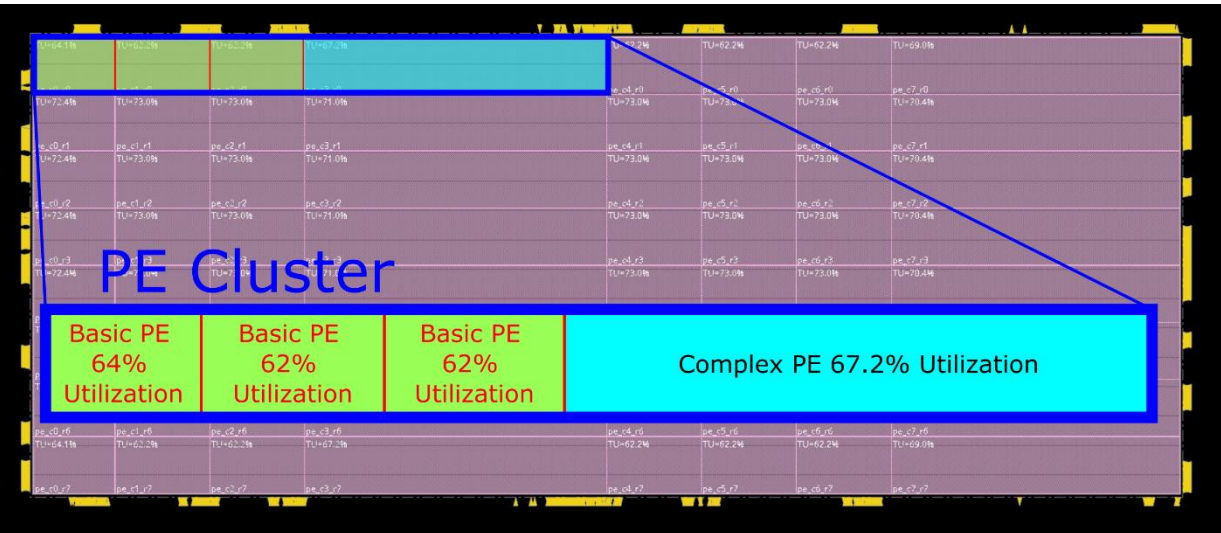
3:1 cluster floorplan



7:1 cluster floorplan

Larger Area for Basic PE in Cluster Floorplan

Lower utilization due to region constraint



3:1
Column floorplan



3:1
Cluster floorplan

Summary



Hiring researchers,
Contact me!

Reconfigurable data-flow computing should be promising for power-efficient HPC.

- **FPGAs are suitable for domain-specific computing.**

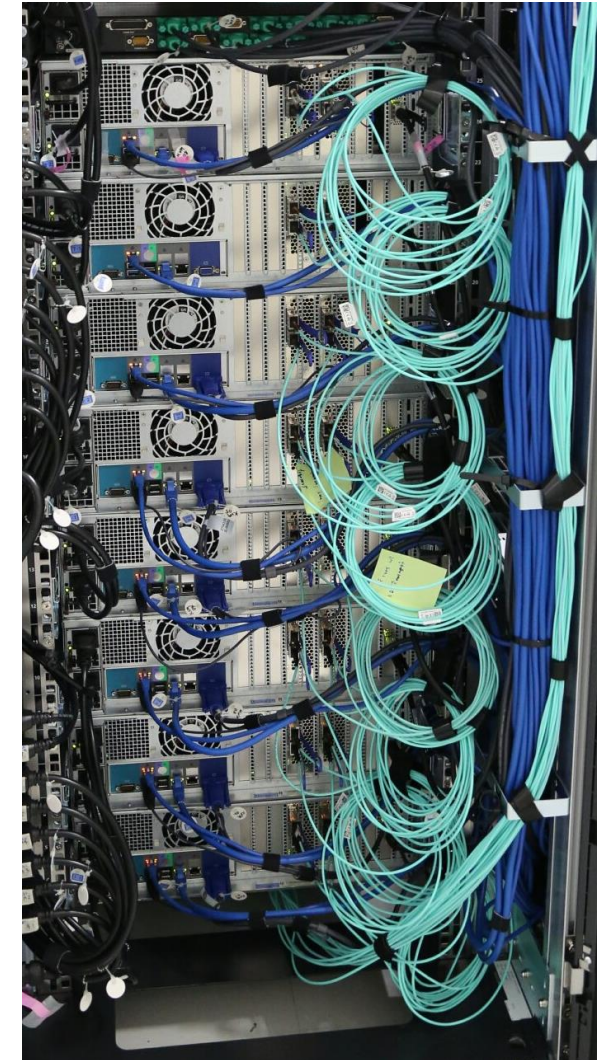
- ✓ *ESSPER: FPGA cluster testbed*
- ✓ *Quantum error correction*

- **CGRA should be better for general HPC.**

- ✓ *RIKEN CGRA for HPC and AI*
- ✓ *Need engineering work and compiler development*

Future work

- ✓ **ESSPER2 with Altera Agilex-M FPGA** (mainly for Quantum research)
- ✓ SoC design of CGRA for HPC and AI (preparation for future ASIC)
- ✓ Have more collaboration!



Thank you !